

Today's Collaboration Software Tools

by John Mohan, CEO, Rosebud LMS, Inc.

Globalization and the cost of travel are clearly in conflict. Both have increased dramatically and are expected to continue to increase for the foreseeable future. As oil dependent travel costs increase, the need to collaborate on the Internet demands better and better software tools. This white paper describes the real world collaboration process and presents a survey of the web software tools that are available today that supports the collaboration process.

Definition of Collaboration:

The collaboration process can be described as a group of people working together to resolve issues and come to agreement on a work product. The group will *independently* review work product documentation, add their comments, respond to comments of the others member and finally reach a consensus. The collaboration process also requires that the group come together *concurrently* so that each member can re-enforce their individual points of view and reach a common agreement. For collaboration to be most effective, *concurrent* meeting is essential. Depending on the work product issues involved, more than one iteration of *independent* and *concurrent* interaction may be required to produce the best and most beneficial result.

The key to effective collaboration is the elaboration of each participant's comments, probing the validity of these comments and the refinement of the comments in relation to everybody else's comments and the overall objective.

Today's collaboration software tools:

There are a wide variety of Internet based software tools available today that try to address the real world collaboration process as described. The simplest of these is email. If there are only two people involved in the collaboration process email *can be* effective. "*Can be*" are the operative words because as email has grown exponentially, it has become less reliable. Occasionally, email can take not only hours but days to get through to the intended party if your email server is down or being hacked which speaks to the popularity of IM today. And if there are more than two people collaborating, email mimics a "ping-pong" effect by requiring everybody's comments to be forwarded to everybody else to keep them in the loop. Even when viewed as threaded email this process is burdensome and unwieldy.

For collaboration to be effective, each person's comments requires feedback from the group and the feedback should offer the original commenter a broader perspective on the issues under consideration. It is precisely this iteration process that produces the best collaborative resolution of the issues. Going beyond email, let's look at the type of web software tools available today that supports the collaboration process.

Collaboration today uses two types of software tools that comprise a bifurcated solution for the real world collaboration process. One prong is the web-meeting tools also referred to as web-conferencing or real time collaboration tools that enable a host or discussion leader to present material live in web-meeting or *concurrent* review by a group. The other prong of this bifurcated solution to real world collaboration is content management tools that supports *independent* review of work product documentation.

Web-meeting tools

Web-meeting tools have evolved from the PC sharing tools of the 1980s like NetMeeting, PCAnywhere and CarbonCopy and the newer versions like RemoteDesktop and VNC that allow one PC to view the screen and share the keyboard/mouse of another PC.

Today's web-meeting tools (i.e. WebEx, MS LiveMeeting, Oracle:iMeeting, IBM:Sametime, Raindance, Cisco:MeetingPlace, Raindance, Web4:eReview, Centra and a host of other vendors) have adopted the same architecture but expanded the number of viewers to be theoretically unlimited given infinite bandwidth. Some of these web-meeting solutions (i.e. MS LiveMeeting) only support Windows platforms and ignore the growing open source Linux market and the persistent Mac market preferred by graphic designers.

These tools capture the screen image of the host or a designated participant, compress the screen image and transmit the compressed image to all other participants. The client software is downloaded to the participants PC that decompresses the transmitted file and displays the image on their platform. The compression schemes used always results in a loss of image quality and resolution; depending on the compression scheme used, you either loose graphic quality (i.e. LiveMeeting) or loose text readability (i.e. WebEx). Bandwidth limitations can sometimes produce partially cross-hatched areas on the viewer's screen where nothing is readable.

Web-meeting tools use either one or both of two methods to deliver live web-meetings. The *first* method, known as "application or desktop sharing", uses a single workstation to run an application to conduct the web-meeting event. Many of the web-meeting vendors use Microsoft PowerPoint as the document manipulation application. As the "host" at the workstation where the application runs directly manipulates a document, each of the remote participants views a dynamically updated screen snapshot of what appears on the host's screen. The remote users can also manipulate the document by having the downloaded client software capture the mouse motion and keyboard entries they make and send these commands to the workstation where the application actually runs.

There are two major shortcomings to this first method: First, it requires large bandwidth to transmit the snapshot across the network and can be slow. This delay can show up on the remote users PC as "cross-hatching" wherein part of the document image is lost. Secondly there is the question of security when the host gives up privacy by exposing his/her platform for all participants to view using desktop sharing. Application/desktop sharing poses a substantial security risk since the technology used

to replicate mouse and keyboard events can also be used to give an unauthorized user control over the host's workstation.

The greatest weakness of using web-meeting tools in the application or desktop sharing mode for collaboration is precisely their greatest virtue: application/desktop sharing. Application/desktop sharing is an absolutely necessary tool to demonstrate a vendor's software products but it is not appropriate for live real time collaboration as some users of these products are just beginning to understand. After you, as the host, have given desktop access to your computer when you are on the web, you have literally installed a "Trojan Horse" on your computer. Ask these vendors: *"Once you have installed their client software on your PC, how do they know that some hacker or disgruntled employee can't access your PC whenever you are on the web without your knowing it?"*

A *second* method used by web-meeting tools, designed to solve the high bandwidth problem of the *first* method, uses a universal document representation scheme and installs on each participant's workstation a client software application to display and edit the document. The entire document is first downloaded after which only changes are transmitted. HTML is most often chosen as the representation scheme and Internet Explorer, which is known to be virus prone, as the web browser for the document manipulation application.

The major disadvantage of using HTML is that Web browsers do not "abstract away" from the workstation-specific issues of screen size and resolution. As a result, the HTML representation is NOT capable of placing a highlighter in the same spot in the document as viewed by all participants in a session. This will cause obvious confusion.

Some vendors (i.e. WebEx and eReview) offers a variant of the second method by capturing a vector graphics (VG) representation of the host's screen in lieu of HTML and a client application that displays the VG representation on the remote platform. To further minimize the bandwidth requirements, the VG representation is usually a lower resolution version of the original document. The lower resolution VG representation will not adequately support a user who wants to zoom in to any part of the document.

One of the other major drawbacks of this scheme is that dynamic links are not preserved in the VG representation. This makes this method unusable for collaborating with web-pages, active links, help files, book-marked files, multimedia content or any dynamic links embedded in the original source documents.

Although this second method used by the Web-meeting vendors can display the comments and annotations made to the document being reviewed, the comments are written on a visual layer on top of the document and the comments are not electronically appended to the document. You can save a recording of the session and some vendors let you save and print a copy of the annotated document but the annotations have to be manually re-entered into your source document.

The deficiencies of both web-meeting tool methods exist because these products were developed for web-presentation and were not designed for document

collaboration. Some of these vendors will actually tell you when you join one of their online demo sessions that their products are not intended for document review, editing, and collaboration.

Content managements tools

Among the vendors that provide these tools are Documentum, OpenText, FileNet , Xerox's DocuShare and MS's SharePoint (a.k.a. Groove). These tools allow an authorized group to independently check-out a documentation for review, add their comments and check the commented documentation back into the server. Today's content management tools are 1970s mainframe and 1980s server based solutions ported to the web. Some tools purport to offer simultaneous (synchronous) co-editing collaboration but what they mean by this is that you can slowly pass control via an editing token from one participant to another; they are nothing more than serial check-in/check-out solutions. None of these tools support truly simultaneous (synchronous) collaboration.

Groove's Virtual Office's is an example of this and supports co-editing for some applications. Groove's trial version 3.1 allows you to co-edit MS Word documents which means than one person at a time can edit a single document and have their changes replicated to the others by saving the document, the changes are then written to the other participants open copies. The person with editing rights can then give another participant editing rights and that person can make a change and save it to let others view it. But only one person may make changes at a time and that person has to be passes the editing token in a slow process before editing. And again this assumes that everybody has the source application.

There are also hybrid products like Web4's eReview that integrates their web-meeting tool into other vendors content management tools and provides a software bridge from one prong (web-meeting) to the other prong (content management). eReview's web-site states that their product is integrated into Documentum, Matrix One, OpenText, Xerox's DocuShare and FileNet. This approach simplifies the transition from one prong to the other but is cumbersome to use and does nothing to address the inadequacies of either or provide a complete solution for the real world collaboration process.

Conclusions about today's solutions

Web-meeting and content management tools are currently trying to incorporate the features of each other but no matter how that is implemented, there is no way of obliterating the deficiencies of the other prong whose architectures are rooted in the 70s and 80s software design. In summary, the two prong solutions of web-meeting and content management are not only awkward but limited in their ability to adequately support the real world collaboration process. And of special future interest, most of these solutions do not provide a growth path to the new cell-phones, PDAs and mobile devices that are beginning to appear.

Both prongs also require that all participants have a copy of the source application that generated the original document. This requirement is prohibitive for most engineering, medical, architectural, automotive and graphic design applications. Requiring all participants to have a common application is also unrealistic when outside contractors and in-house staff are jointly producing a work product. In addition, requiring that all participants have the source application also assumes that the collaboration host/leader accepts direct changes to the source documents. Although I am sure that there are people who find this acceptable, I think that most people prefer to make their own final changes to a source document.

There are exceptions to this requirement like WebEx and eReview whose users can convert a source document to a virtual printer format that is then distributed to all participants. Each participant is downloaded a client viewer that converts this format to a screen image for display on their workstation. And although each participant can see and annotate the document, these annotations are applied to a visual layer on top of the document and each participant's annotations are saved in a separate file on the collaboration server. This can be a very slow annotation process. And although the annotated document can be printed locally, some solutions don't allow you to save it locally for offline review and, more importantly, the annotations cannot be round-tripped back to the source application (i.e. MS Word or AutoCAD) that created the document.

Now let's examine what happens when everybody has added their comments and the group host/leader harvests all the annotations, accepts some, rejects others and schedules a teleconference to resolve the open issues. This requires that the host either assume that everybody has the source application or convert the annotated documentation to a viewable format for the web-meeting tool. Aside from the loss of resolution, this is a practically untenable process.

Another important issue that needs to be considered when selecting a collaborative tool is the degree of user training required to integrate these products into your environment. Some of these tools are quite difficult for a user to become familiar with and a few of them seem to be targeting professional programmers as users.

Rosebud's breakthrough solutions:

Rosebud was conceived and designed to avoid the short-comings of web-meeting and content management tools and to address eight (8) requirements for real world collaboration that have not previously been integrated into a Web software solution that:

- Can be used with any document type.
- Preserves the content and quality of the original document.
- Works the same for both independent and concurrent collaboration.
- Offers multiple levels of document security.

- Is platform independent.
- Uses a well-known, widely distributed, easy-to-use application for document manipulation to minimize user training.
- Retains the *dynamic* capability of electronic documents.
- Alerts offline members of online collaboration team activity.

To meet these requirements, Rosebud uses:

- o PDF, the de-facto standard for electronic content for document representation.
- o Java for optimum security and cross platform independence.
- o And our patent pending software architecture that puts it all together to support independent and concurrent collaboration by eliminating the distinction between them.

[Rosebud](#) uses "PDF" with its inherently greater image resolution and preservation of the source document as the document representation scheme and Adobe Acrobat that does not rely on dangerous software features as the document manipulation application. Visit our website to learn more and download the free trial version of Rosebud: www.Rosebudplm.com